

CLAIMS

What is claimed is:

- 1 1. A method of rendering a graphics primitive to a frame buffer for a display, the frame buffer
2 being partitioned into a plurality of equal size blocks of pixels, each block having at least two
3 depth-layers, the frame buffer having an associated z-buffer with an depth value for each pixel in
4 the frame buffer, the depth value determining the visibility of a pixel on the display, the method
5 comprising:
 - 6 determining the blocks in the frame buffer that are affected by the graphics primitive;
7 for each affected block,
 - 8 adjusting the depth layers of the block,
9 determining whether any pixels of the graphics primitive for the block intersect
10 any of the depth-layers of the block, the block being partitioned into a plurality of equal size sub-
11 blocks and having a sub-block mask that includes a bit for each sub-block; and
12 if any pixels of the graphics primitive for the block intersect any of the depth-
13 layers, setting a flag indicating that the z-buffer requires an update for the block, forming a pixel
14 mask that indicates which pixels in the block are intersecting pixels, and combining the pixel
15 mask with the sub-block mask to determine which sub-blocks have intersecting pixels;
16 again determining the blocks in the frame buffer that are affected by the graphics
17 primitive; and
18 for each affected block,
 - 19 adjusting the depth layers of the block,
20 determining whether or not there is an update flag set for the block,
21 if there is no update flag set, rendering the visible pixels to the frame buffer based
22 on the adjusted depth layers
23 if the update flag is set, then for each sub-block of the affected block, determining
24 based on the sub-block mask whether the sub-block is affected by any intersecting pixels; and if
25 the sub-block is affected, reading the z-buffer, adjusting the visibility entry for the intersecting
26 pixels, and rendering any visible pixels to the frame buffer.

1 2. A method of rendering a graphics primitive to a frame buffer, as recited in claim 1,
2 wherein there is a front depth-layer and a back-depth layer for each affected block, and
3 the front and back layers each have a range of depth values in the block, each range having a min
4 and max value, the min value being the closer value; and
5 wherein the graphics primitive has a range of depth values for the pixels affected in the
6 block by the primitive, the graphics primitive range having a min and max value.

1 3. A method of rendering a graphics primitive to a frame buffer, as recited in claim 2, wherein a
2 mask is associated with the block, the mask describing which pixels in the block belong to the
3 front layer and to the back layer.

1 4. A method of rendering a graphics primitive to a frame buffer, as recited in claim 2, wherein a
2 mask is associated with the graphics primitive, the mask describing which pixels in the block are
3 affected by the graphics primitive.

1 5. A method of rendering a graphics primitive to a frame buffer, as recited in claim 2,
2 wherein pixels affected in the block by the graphics primitive intersect both the front
3 layer and the back layer;
4 wherein the pixel mask indicates that all the pixels affected in the block are intersecting
5 pixels;
6 wherein the front and back layer are both adjusted to have a range of depth values
7 between
8 (i) the minimum of the graphics primitive min and front layer min and
9 (ii) the maximum of the graphics primitive max and back layer max, and
10 to include front layer, back layer and pixels affected in the block by the graphics
11 primitive.

1 6. A method of rendering a graphics primitive to a frame buffer, as recited in claim 2,
2 wherein pixels affected in the block by the graphics primitive intersect the front layer but
3 not the back layer;
4 wherein the pixel mask indicates which of the front layer pixels are intersected;

5 wherein if the pixels affected in the block cover all of the back layer in the block, the
6 front and back layers are both adjusted to have a range of depth values between

7 (i) the minimum of the graphics primitive min and the front layer min,
8 (ii) the maximum of the graphics primitive max and front layer max, and
9 to include the front layer pixels and the pixels affected in the block by the
10 graphics primitive;

11 wherein if the pixels affected in the block do not cover all of the back layer, the front
12 layer is adjusted to have a range of depth values between

13 (i) the minimum of the graphics primitive min and the front layer min,
14 (ii) the maximum of the graphics primitive max and the front layer max, and
15 to include in the front layer the pixels affected by the graphics primitive and the
16 front layer pixels; and

17 the back layer range is unchanged and includes only pixels not covered by the
18 pixels affected by the graphics primitive.

1 7. A method of rendering a graphics primitive to a frame buffer, as recited in claim 2,

2 wherein pixels affected in the block by the graphics primitive intersect the back layer but
3 not the front layer and the front layer does not cover all of the pixels affected by the graphics
4 primitive;

5 wherein the pixel mask indicates which pixels of the back layer are intersected;

6 wherein the back layer is adjusted to have a range between

7 (i) the minimum of the graphics primitive min and the back layer min,
8 (ii) the maximum of the graphics primitive max and the back layer max, and
9 to include in the back layer all of the pixels in the back layer affected by the
10 graphics primitive; and

11 wherein the front layer range is unchanged.

1 8. A graphics rendering apparatus comprising;

2 a frame buffer for storing pixels to be displayed, the frame buffer being partitioned into a
3 plurality of equal size blocks of pixels, each block being partitioned into a plurality of equal size
4 sub-blocks;

5 a z-buffer associated with the frame buffer, the z-buffer storing a depth value for each
6 pixel in the frame buffer, the depth value determining the visibility of a pixel on the display; and
7 a graphics processor for processing the pixels of the z-buffer and frame buffer, the
8 graphics processor including
9 z-range buffer having fields for indicating for a block a depth range for a front layer, a
10 depth range for a back layer, a mask indicting which pixels in the block are in the front layer and
11 back layer; and a field for storing sub-block information that indicates whether or not the z-
12 buffer must be read for the sub-block;
13 a first register having fields for indicating which pixels in a block are affected by a
14 graphics primitive and for indicating the depth range of the pixels in the block affected by the
15 graphics primitive;
16 a second register having a field for indicating which pixels in the block require the z-
17 buffer to be read, and
18 a flag indicating that the z-buffer must be read for a block.

1 9. A graphics rendering apparatus as recited in claim 8, wherein the graphics processor is
2 configured to:

3 determine which pixels in a block are affected by a graphics primitive and the depth
4 range of the pixels in a block affected by the graphics primitive and to set the fields in the first
5 register accordingly;

6 determine whether the pixels in a block affected by a graphics primitive intersect the
7 front layer, the back layer or both layers;

8 set the flag if the z-buffer must be read for a block;

9 adjust the ranges of the front and back layers in the z-range buffer based on the pixels in a
10 block affected by the graphics primitive and the depth range of the affected pixels;

11 if the flag is not set, render visible pixels to the frame buffer depending on the adjusted
12 front and back layers;

13 if the flag is set, determine which pixels in the block require the z-buffer to be read and
14 setting the field in the second register accordingly based on the intersection determination,
15 determine which sub-blocks in the block require the z-buffer to be read and set the field in the z-
16 range buffer accordingly, read the z-buffer to obtain the depth values for the sub-block, and

17 render the visible pixels to the frame buffer depending on the depth values obtained for the sub-
18 block.